

AD-A227 189

NASA Contractor Report 182074

ICASE Report No. 90-51

ICASE

LINEAR ITERATIVE SOLVERS FOR IMPLICIT ODE METHODS

Paul E. Saylor
Robert D. Skeel

Contract No. NAS1-18605
August 1990

Institute for Computer Applications in Science and Engineering
NASA Langley Research Center
Hampton, Virginia 23665-5225

Operated by the Universities Space Research Association

DTIC
ELECTE
OCT 03 1990
S E D
6

NASA

National Aeronautics and
Space Administration

Langley Research Center
Hampton, Virginia 23665-5225

DISTRIBUTION STATEMENT A

Approved for public release;
Distribution Unlimited

90 10 08 049

~~90 10 08 049~~

LINEAR ITERATIVE SOLVERS FOR IMPLICIT ODE METHODS¹

Paul E. Saylor

Robert D. Skeel

Department of Computer Science
University of Illinois at Urbana-Champaign
240 Digital Computer Laboratory
1304 West Springfield Avenue
Urbana, Illinois 61801-2987

Accession For	
DTIC	<input checked="checked" type="checkbox"/>
NTIS	<input type="checkbox"/>
Un	<input type="checkbox"/>
Distribution/	
Availability Codes	
Disc	Full and/or Special

A-1

ABSTRACT

In this paper we consider the numerical solution of stiff initial value problems, which lead to the problem of solving large systems of mildly nonlinear equations. For many problems derived from engineering and science, a solution is possible only with methods derived from iterative linear equation solvers. A common approach to solving the nonlinear equations is to employ an approximate solution obtained from an explicit method. In this paper we shall examine the error to determine how it is distributed among the stiff and non-stiff components, which bears on the choice of an iterative method. Our conclusion is that error is (roughly) uniformly distributed, a fact that suggests the Chebyshev method (and the accompanying Manteuffel adaptive parameter algorithm). We describe this method, also commenting on Richardson's method and its advantages for large problems. We then apply Richardson's method and the Chebyshev method with the Manteuffel algorithm to the solution of the nonlinear equations by Newton's method.

¹Work supported in part by the National Science Foundation Grant DMS 89 11410 and by the US Department of Energy grant DOE DEFG02-87ER25026. The research for the first author was partially supported by the National Aeronautics and Space Administration under NASA Contract No. NAS1-18605 while in residence at the Institute for Computer Applications in Science and Engineering (ICASE), NASA Langley Research Center, Hampton, VA 23665.

1 Introduction

The numerical solution of initial value problems requires at each time-step the solution of an implicit nonlinear equation, usually by a Newton-like iteration. As a result it is necessary to solve a system of linear equations, which is often large and sparse. It has been customary to use direct methods, but in recent years studies on the suitability of iterative methods [18, 9, 3, 1] have been emerging.

This paper examines some issues in the application of iterative methods to initial value problems (IVPs). *First*, we identify more clearly the nature of the task to be performed by the iterative solver. A number of comments have been made about the question of whether the predictor error lies primarily in the dominant subspace. These comments have not been convincingly supported by analysis. The difficulty with the analysis of stiff equations is that of knowing which quantities are large and which are small — these things change during the course of the integration. We believe we have found a satisfactory way of dealing with this question and that is to regard as small those quantities which the error control mechanisms of the algorithm make small. *Second*, we explain why the Chebyshev method might be well suited as an inner iteration to solve the linear equations arising at each step of the Newton step. In addition we consider an application of the first order Chebyshev method to the outer, Newton iteration. (The first order Chebyshev method is Richardson's method with Chebyshev parameters.)

This paper is a preliminary theoretical study; the accuracy and usefulness of our observations await experimental confirmation.

1.1 Outline of the Paper

In §2, the standard time-stepping approach for solving linear and nonlinear stiff IVPs is described in a simplified way. When the IVP is nonlinear, a variant of Newton's method is employed at each time step, usually the modified Newton's method in which the Jacobian is not always up-to-date. (It will be convenient to say Newton's method, however, rather than either modified Newton's method or, as is also used, the modified inexact Newton's method.) Each Newton step requires the solution of a linear system for which the matrix is a Jacobian, either the current Jacobian or one saved from a previous Newton step. (Matrix-free methods avoid explicit computation of the Jacobian but from time to time the Jacobian will

be referred to as though it had been computed and is available.) Assume that an iterative method is used to solve the linear system. This is an inner iteration whereas Newton's method is an outer iteration in a combined inner-outer iteration.

Iterative methods differ in the ways in which the error is reduced. In §3, the error in the predictor step is analyzed. The predictor step may be thought of as the initial guess for the inner iteration, leading to the discussion in §4 of appropriate (inner) iterative methods for reducing the error in the predictor.

In §5, an application of techniques employed in the Chebyshev iterative method is made to the outer iteration to give more control over convergence.

1.2 Terminology and Notation

It is customary to discuss the solution of IVPs only in the real case. However, to take into account the importance of complex IVPs, we use the more general and more appropriate terms *hermitian* and *nonhermitian* rather than symmetric and nonsymmetric.

The Jacobian of functions f and F will be denoted by f' and F' respectively.

The computed approximation to the solution of an IVP at step n will be denoted by y_n . Usually, y_n is obtained by approximately solving a nonlinear equation, the exact solution of which, when clarity is required, will be denoted by y_n^* .

2 Nonlinear Iteration in IVPs

Assume a transient problem $y'(t) = f(y(t))$ with an appropriate set of initial values. (For convenience, we assume the system is expressed in autonomous form.) If the equation is stiff, the standard numerical solution methods, based on Newton's method, yield linear systems to be solved. We illustrate with backward Euler. There is little reason to believe that the ideas do not apply to other implicit difference schemes. For backward Euler, the approximation, y_n , to $y(t_n)$ at t_n is the computed solution of

$$y_n^* = y_{n-1} + h_n f(y_n^*) \quad (1)$$

This is an *implicit equation* for the unknown y_n^* . Rearranging (1) slightly, it is necessary to solve an equation of the form

$$F(y_n^*) := \frac{1}{h_n} y_n^* - f(y_n^*) - \frac{1}{h_n} y_{n-1} = 0 \quad (2)$$

The solution of (2) by Newton's method requires the solution of the linear systems

$$F'(y_n^{(m)})(y_n^{(m+1)} - y_n^{(m)}) = -F(y_n^{(m)}). \quad (3)$$

Customarily using a direct method one iterates with

$$y_n^{(m+1)} = y_n^{(m)} - c\hat{G}^{-1}F(y_n^{(m)}) \quad (4)$$

where \hat{G} is a Jacobian of F kept either from an old time step or from an earlier iteration and c is an acceleration parameter. Since \hat{G} is a matrix, it follows that computing the vector $x = \hat{G}^{-1}F(y_n^{(m)})$ is equivalent to solving a set of linear equations,

$$\hat{G}x = F(y_n^{(m)}). \quad (5)$$

The solution of the nonlinear equation (2) is called the *corrector* step, and the initial approximation of $y_n^{(0)} := y_n^p$ obtained from an explicit multistep formula is called the *predictor* step.

One could instead approximate the solution of (3) by means of a matrix-free iterative method. Such methods require only that we be able to compute the action of the Jacobian matrix $F'(y_n^{(m)})v$ for an arbitrary vector v . This can be accurately approximated by

$$\frac{1}{\delta} [F(y_n^{(m)} + \delta v) - F(y_n^{(m)})]$$

for some small δ of the order of the square root of the machine epsilon. Note that $F(y_n^{(m)})$ is already available, because it is the right-hand side of the linear system (3). Hence each iteration of a typical linear iterative solver requires but one function evaluation. The stopping criterion for the inner linear iteration would be based on that of the outer Newton iteration.

This question of when to stop the inner iteration is not easy: one would like to avoid unnecessary inner iterations without degrading the convergence of the outer iteration. Hence it has been proposed [4] that a single nonlinear iteration would be more efficient than a two-level iteration. However the one-level iteration proposed in [4] requires two function evaluations per iteration; whereas the two-level iteration requires one function evaluation per inner iteration and one per outer iteration.

Currently it seems more practical to use iterative methods to solve (5) if there is some preconditioning, although this usually requires that an explicit matrix be available.

3 Accuracy Needed

Much discussion has focused on this objective of the nonlinear solution to impose stability [9, 3]. In this section, we present some analysis to show what components of the error should be reduced by an iterative method.

Consider now the backward Euler formula

$$y_n = y_{n-1} + h_n f(y_n) + h_n F_n \quad (6)$$

where the residual (per unit step) $F_n := F(y_n)$ would be zero if the nonlinear system were solved exactly.

One question concerns the nature of the error in the predictor. In practice, the predictor is usually

$$y_n^p = y_{n-1} + h_n y_{n-1,n-2} \quad (7)$$

where the double subscript denotes a first divided difference. (Often this is written $y_n^p = y_{n-1} + h_n y'_{n-1}$ where y'_{n-1} is chosen, *not* to satisfy $y'_{n-1} = f(y_{n-1})$, but rather to satisfy the formula used on the $(n-1)$ -st step. If this formula is backward Euler, then $y_{n-1} = y_{n-2} + h_{n-1} y'_{n-1}$ determines y'_{n-1} .) The residual $F_n^p := F(y_n^p)$ for the predicted value satisfies

$$y_n^p = y_{n-1} + h_n f(y_n^p) + h_n F_n^p \quad (8)$$

and the nature of this residual depends on how the stepsize is controlled. It is customary to control the stepsize h_{n-1} so that $\|\tilde{l}e_{n-1}\| \leq \epsilon$ and to choose h_n so that $r_n^2 \|\tilde{l}e_{n-1}\| \leq$ (safety factor) ϵ , where ϵ is the local error tolerance, $\tilde{l}e_{n-1}$ is a *local error estimate*, and r_n is the stepsize ratio h_n/h_{n-1} . For the local error estimate, it is common to use

$$\tilde{l}e_n = h_n^2 y_{n,n-1,n-2} \quad (9)$$

where the triple subscript denotes a second divided difference. What is wanted is an expression for the predictor residual which, as much as possible, is in terms of quantities known to be small. It will be shown at the end of this section that

$$\begin{aligned} F_n^p &= (1 + r_n)F_{n-1} - r_n F_{n-2} \\ &\quad - \frac{1}{h_n} r_n^2 (1 + r_{n-1}^{-1}) \tilde{l}e_{n-1} + \text{nonlinear term} \end{aligned} \quad (10)$$

where the nonlinear term is given in a later paragraph.

Let y_n^* be the exact solution of $F(y_n^*) = 0$. Codes attempt to control the error, $\delta_n = y_n^* - y_n$, in the solution typically so that it is no greater than some fraction ϕ of the local error tolerance:

$$\|\delta_n\| \leq \phi \epsilon (\text{hopefully}).$$

In practice, an approximation for the error is necessary, and usually this is a scalar multiple of $y_n^{(m+1)} - y_n^{(m)}$. The widely distributed package DASSL[20] uses $\phi = 1/3$. We now want to look at (10) to determine what the iterative linear equation solver must do. To simplify this description, neglect the nonlinear term as well as variations in the stepsize and in the Jacobian matrix f' of f . Under these assumptions the error δ_n in solving for y_n satisfies

$$\delta_n = - \left(\frac{1}{h} I - f' \right)^{-1} F_n$$

and the predictor error δ_n^p satisfies

$$\delta_n^p = 2\delta_{n-1} - \delta_{n-2} + 2(I - hf')^{-1} \tilde{e}_{n-1}. \quad (11)$$

The conclusion is that the nonstiff error must be reduced by as much as $1/(3 + 2\phi^{-1})$ (set $hf' = 0$) and the stiff error by as much as $1/3$ (set $hf' = -\infty$). Trying to control the error in an iteration based on the size of the correction is tricky unless the convergence is rapid; thus, DASSL requires a convergence factor not greater than 0.9. Therefore, for an iterative method one might want to consider controlling the residual instead [25, 3]. The price for this is that instead of reducing the error by $1/(3 + \phi^{-1})$ in just the nonstiff components, the more severe reduction must be done for all components. In addition, [5] argues that the possible skewness of the eigensystem of f' can make it risky to control the residual.

One might consider loosening up on the fraction ϕ ; however, the stability of the method [16] and the reliability of the local error estimation depends on having an accurate solution of the nonlinear system. Nonetheless, this matter should be re-examined because of its importance to the efficient use of iterative (non)linear equation solvers.

It remains to discuss the nonlinear term in (10), which can be shown to be

$$\text{nonlinear term} = \frac{1}{2} h_n^2 (1 + r_n^{-1}) (\bar{f}'')_{n-1} y_{n-1,n-2} y_{n-1,n-2}$$

where $(\bar{f}'')_{n-1}$ is some average value of f'' . It seems quite possible that this could be a significant part of the stiff error for some problems. It can be shown that the contribution of the nonlinear term to the residual can be computed at a cost of one function evaluation (and

estimated for practically nothing). It is a question whether the nonlinear error has a special structure, and whether stepsize control should ensure that the nonlinear error is small.

Recently, a different stepsize control strategy has been proposed in [14], which for our example would mean using $h_n y_{n,n-1}$ for the local error. This is consistent with current practice in nonstiff solvers where the stepsize is controlled for a formula that is one order lower than that actually used. The foregoing analysis of the predictor error simplifies in this case.

We conclude this section by deriving (10). Using Eqs. $(6_{n-2})^2$, (6_{n-1}) , and (8_n) to determine F_{n-2} , F_{n-1} , and F_n^P with y_n^P eliminated by Eq. (7), we get

$$F_n^P - F_{n-1} = z(t_n) - z(t_{n-1}) \quad (12)$$

and

$$F_{n-1} - F_{n-2} = z(t_{n-1}) - z(t_{n-2}) - (y_{n-1,n-2} - y_{n-2,n-3}) \quad (13)$$

where

$$z(t) = f(y_{n-1} + (t - t_{n-1})y_{n-1,n-2}).$$

Hence, using (12) and (13) to create second divided differences, we get

$$F_{n,n-1,n-2}^P = z[t_n, t_{n-1}, t_{n-2}] - (h_n + h_{n-1})^{-1}(1 + r_{n-1}^{-1})y_{n-1,n-2,n-3}.$$

We can express the divided difference

$$z[t_n, t_{n-1}, t_{n-2}] = \int_{t_{n-2}}^{t_n} K(t) z''(t) dt$$

where the Peano kernel $K(t)$ is the hat function for $t = t_{n-1}$ divided by $h_n + h_{n-1}$. If we define

$$(\overline{f_{vv}})_{n-1} = 2 \int_{t_{n-2}}^{t_n} K(t) f_{vv}(y_{n-1} + (t - t_{n-1})y_{n-1,n-2}) dt$$

and use (9), we get

$$F_{n,n-1,n-2}^P = \frac{1}{2}(\overline{f_{vv}})_{n-1} y_{n-1,n-2} y_{n-1,n-2} - (h_n + h_{n-1})^{-1}(1 + r_{n-1}^{-1}) h_{n-1}^{-2} \tilde{e}_{n-1}$$

from which (10) follows.

²The notation means that in (6, n is replaced by $n - 2$, and similarly in the next two references.

4 Iterative Linear Solution Methods

In this section, iterative methods will be sketched, focusing on the Chebyshev method and Richardson's method. Technicalities will not be emphasized.

4.1 Krylov Subspace Methods

Let $Ax = b$ be a linear set such as in (3). Let $x^{(0)}$ be an initial guess and $r^{(0)} := b - Ax^{(0)}$ be the initial residual. The iterative methods that will be considered here are such that the solution approximation is of the form $x^{(i)} = x^{(0)} + y$, where $y \in V_i := \text{span}\{r^{(0)}, \dots, A^{i-1}r^{(0)}\}$, and V_i is the *Krylov subspace*. Such methods are called *Krylov subspace methods*. (Another widely used term for Krylov subspace methods is *polynomial methods*.) They include and unify a large class of methods, such as the conjugate gradient method, and for nonhermitian matrices, the adaptive Chebyshev iteration of Manteuffel [17], and Richardson's method together with GMRES [21], ORTHOMIN and other conjugate gradient-like methods.

4.2 The Residual Polynomial

Let $x^{(i)}$ be the i th approximation resulting from an iterative method. Also let $e^{(i)} = x - x^{(i)}$ and $r^{(i)} = b - Ax^{(i)}$ be the *error* and *residual* respectively. For a Krylov subspace method, the error $e^{(i)}$ is related to the initial error by a *residual polynomial* R_i defined by $e^{(i)} = R_i(A)e^{(0)}$. The name results from the fact that the same polynomial propagates the residual, $r^{(i)} = R_i(A)r^{(0)}$. (The GMRES and ORTHOMIN methods are restart methods for which $e^{(0)}$ may be the initial error due not to the initial approximation of $Ax = b$ but to some later approximation. Also the adaptive Chebyshev method of Manteuffel [17] is a restart method.)

4.3 Types of Acceleration

Methods differ in the way that the residual polynomial is determined. Thus, in conjugate-gradient-like methods the residual polynomial minimizes a certain weighted norm of the error. In the *Chebyshev method* the polynomial is chosen as that polynomial minimizing a uniform norm over an ellipse among all residual polynomials of a fixed degree and turns out to be a scaled and translated Chebyshev polynomial. There are practical differences among these methods of course. Conjugate-gradient-like methods determine their residual polynomial

in an automatic way whereas the Chebyshev method requires a set of parameters, which, nevertheless, can be computed in an effectively automatic way by means of the Manteuffel algorithm [17].

The Manteuffel algorithm chooses parameters based on an estimate of the convex hull³ of the eigenvalues, beginning, for example, with a point known to be inside the convex hull. After every 20 or so iterations the convex hull is expanded (and the parameters recomputed) to include estimates of several stray eigenvalues. The combination of the Chebyshev method with the Manteuffel algorithm will be referred to as the *adaptive Chebyshev method (of Manteuffel)*.

One advantage of the adaptive Chebyshev method is as follows. The solution of stiff IVPs yields a succession of linear systems to solve, each one often not much different from the previously solved problem. This is certainly true for the linear systems arising from successive Newton iterations. From one time-step to the next there may be a significant change in $A = \frac{1}{h}I - f'$ due to changes in h , but it is straightforward, in the absence of preconditioning, to calculate what this does to the eigenvalues of the matrix A . The idea then is that eigenvalue estimates from one linear system can be used as initial estimates for the next system. However, because the algorithm works only by expanding the convex hull approximation, it would be important to begin the solution of a new linear system by suitably shrinking the final convex hull from the previous system so that it lies inside the true convex hull.

4.4 Richardson's Method

The Chebyshev method minimizes the residual polynomial over an ellipse, which will be called the *Manteuffel ellipse*, containing the convex hull (approximately) of the spectrum of the system matrix A . If the convex hull is not well-approximated by the enclosing Manteuffel ellipse, then one might want to consider an iterative method for which the residual polynomial is minimized over the convex hull. If the convex hull is not elliptical a *second order iteration* is not, in general, possible. (A second order iteration is one for which the iterate is optimum at each step; the Chebyshev iteration is an example. Second order iterations require that the residual polynomial be generated by a three term recursion. In general a three term recursion does not exist.) However, a *first order method* is always possible. A first order

³The *convex hull* of a set of points is the intersection of the convex sets containing the set of points.

method is often called *Richardson's method*. (For a statement of Richardson's method in algorithm form, which is not necessary for this discussion, see [13].)

It should be noted that Richardson's method does not require working with a convex hull; a general nonconvex set containing the eigenvalues may be used instead. A reason for using the convex hull is that it may be computed from known procedures, whereas there is no known, general procedure for computing a non-convex region containing the spectrum.

If Richardson's method is used, the parameter difficulty is much the same as for the adaptive Chebyshev algorithm of Manteuffel. In fact if the residual polynomial is a scaled and translated Chebyshev polynomial, then Richardson's method is just a first order implementation of the second order Chebyshev method used in the Manteuffel algorithm.

In [23], an adaptive algorithm is presented for Richardson's method that determines the convex hull of the spectrum in a way analogous to the Manteuffel algorithm. It should be noted that the adaptive method in [23] is a combination of two iterative styles, one from the GMRES method, used both to advance the solution and compute an approximate (convex hull of the) spectrum, and the other from Richardson's method. This combination was suggested in [6].

One final note on Richardson's method: The simplicity of the method is an advantage on advanced processors when organizing the computations to minimize data traffic [22].

4.4.1 Minimizing the L_2 Norm of Residual Polynomials

If the convex hull is determined, then one can determine the residual polynomial in an optimum way to satisfy a weighted L_2 norm induced from the inner product,

$$(g, h)_w = \int_{\Gamma} g(\lambda)h(\lambda)w(\lambda)|d\lambda| \quad (14)$$

where Γ is a contour in the complex plane, the boundary of the convex hull of the spectrum in the application to the Richardson's method parameter problem.

The L_2 *optimal residual polynomial* of degree k is defined to be the solution of the least squares problem $(R_k, R_k)_w = \text{minimum}$. There are methods [24] to compute the roots of this polynomial related to the stable algorithm of Golub and Welsch for the computation of nodes and weights for Gaussian quadrature [11]. The reciprocals of the roots of the polynomial then become the parameters required for Richardson's method.

4.4.2 Minimizing the Uniform Norm of Residual Polynomials

In the literature recently there have been ideas developed for the practical computation of Richardson's method parameter for which the residual polynomial is approximately minimized in the uniform norm [7, 8, 19, 26]. These methods use the theory of conformal mapping and the properties of Faber polynomials. The resulting residual polynomial may be called here the L_∞ optimal residual polynomial. An important contribution from [8, 26] is an algorithm for ordering the parameters for Richardson's method to ensure stability.

In the adaptive algorithm in [23], an L_2 optimal residual polynomial is computed and used but this is not a restriction and the L_∞ optimal residual polynomial could be used instead.

4.5 The Appropriate Iterative Method for IVPs

The nature of the error reduction needed to solve the IVP was briefly explained in §3. Equation (11) shows that

$$\delta_n^p = e' + A^{-1} e''$$

where $A = I - hf'$, $\|e'\| \leq 3\phi\epsilon$ and $\|e''\| \leq 2\epsilon$. From this expression, one sees that the nonstiff error must be reduced by as much as $1/(3 + 2\phi^{-1})$ and the stiff error by as much as $1/3$. A simplified and satisfactory conclusion to draw from this is that the stiff and nonstiff errors should be damped uniformly. Thus if the spectrum is well approximated by an ellipse, the adaptive Chebyshev method of Manteuffel is reasonable. If it is not, one may want to consider Richardson's method. If the residual polynomial is determined by minimizing the L_2 norm of the residual polynomial derived from the inner product (14), then a reasonable choice for the weight is

$$w(\lambda) = 1.$$

It should be noted that this is not the Chebyshev weight function, which is the weight function that should be selected if the convex hull is an interval (of positive numbers), and so causes the L_2 norm to behave like the uniform norm only approximately.

In using either the conjugate gradient method in the hermitian positive definite case or conjugate gradient-like methods in the nonhermitian case a weight is imposed on eigencomponents of the error equal to the magnitude of the corresponding eigenvalue. As argued here, such a weight function is not appropriate.

5 Accelerating the Newton Iteration

From §2, the Newton iteration is, for $m \geq 0$,

$$y_n^{(m+1)} = y_n^{(m)} - c\hat{G}^{-1}F(y_n^{(m)}) \quad (15)$$

where c is an acceleration parameter and

$$F(y) := \frac{1}{h_n}y - f(y) - \frac{1}{h_n}y_{n-1}. \quad (16)$$

(Recall that the solution of $F(y) = 0$ is denoted by $y = y_n^*$.) A sequence of acceleration parameters will be derived by interpreting iteration (15) as a preconditioned Richardson's iteration. With an assumption of linearity, acceleration parameters may then be derived by a straightforward application of standard adaptive iteration parameter techniques [17, 13, 23]. Other approaches are possible, for example, [15]. An extended treatment of the solution of nonlinear equation by Krylov subspace methods is given in [2].

5.1 Krylov Subspace from a Nonlinear Operator

A subspace will be defined that is generated by the *preconditioned* nonlinear operator, $\hat{G}^{-1}F$, appearing in Newton's method. A linear approximation to the nonlinear operator allows the subspace to be interpreted as a Krylov subspace. In turn, the Krylov subspace yields acceleration parameters based on Chebyshev polynomials.

5.1.1 Linear Approximation

Let M be the Jacobian of $\hat{G}^{-1}F$ evaluated at $y_n^{(0)}$, $M = \hat{G}^{-1}F'(y_n^{(0)})$. Matrix M is only an aid to exposition, and, of course, is never computed. The true assumption is not that the Jacobian is evaluated at $y_n^{(0)}$ but that the Jacobian is slowly varying during some portion of the Newton iteration. If so, then the point at which M is said to be the Jacobian of $\hat{G}^{-1}F$ is arbitrary, and, in particular, the choice $y_n^{(0)}$ is arbitrary.

Since \hat{G} is the Jacobian of F evaluated at some $y_{n'}^{(m')}$, it would follow that $M \neq I$ unless $y_{n'}^{(m')} = y_n^{(0)}$. Therefore, assume that $y_{n'}^{(m')} \neq y_n^{(0)}$.

5.1.2 Newton's Method as Richardson's Method

Given an approximation $y_n^{(m)}$ to the solution of $\hat{G}^{-1}F(y_n) = 0$, let

$$r^{(m)} = -\hat{G}^{-1}F(y_n^{(m)})$$

be the residual due to $y_n^{(m)}$. Equation (15) becomes

$$y_n^{(m)} = y_n^{(m-1)} + cr^{(m-1)}. \quad (17)$$

Let

$$e^{(m)} = y_n^* - y_n^{(m)}.$$

Subtract (17) from $y_n^* = y_n^*$ to obtain

$$e^{(m)} = e^{(m-1)} - cr^{(m-1)}. \quad (18)$$

Next, since

$$r^{(m)} = \hat{G}^{-1}F(y_n^*) - \hat{G}^{-1}F(y_n^{(m)}) \approx Me^{(m)}$$

it follows from (18) that

$$e^{(m)} \approx (I - cM)e^{(m-1)}. \quad (19)$$

Therefore

$$e^{(m)} \approx (I - cM)^m e^{(0)}. \quad (20)$$

Multiply (20) by M to obtain

$$r^{(m)} \approx (I - cM)^m r^{(0)}. \quad (21)$$

5.1.3 Krylov Subspace

As the Newton iteration proceeds, a Krylov subspace, V_k , is defined by $I - cM$

$$V_k := \text{span}\{r^{(0)}, \dots, (I - cM)^{k-1}r^{(0)}\}.$$

Although V_k is not computed, the subspace

$$\mathcal{V}_k := \text{span}\{r^{(i)}\}_{i=0}^{k-1}$$

is available and

$$V_k \approx \mathcal{V}_k.$$

Computations requiring V_k can be performed using \mathcal{V}_k .

5.2 Chebyshev Acceleration Parameters

From the k vectors in \mathcal{V}_k , $k - 1$ eigenvalues of M may be estimated. (It should be noted that in practice, k is a small number; in the code DASSL, the total number of Newton steps is limited to 4.) From these eigenvalues, the convex hull of the spectrum of $I - cM$ can be computed, which yields [17] two parameters d and c_{ellipse} from which a sequence of acceleration parameters c may be computed. Parameter d is the center of a family of confocal ellipses for which the foci are $d \pm c_{\text{ellipse}}$. The Manteuffel algorithm determines these parameters in order to optimize convergence of the Chebyshev method.

The d and c_{ellipse} parameters yield a sequence of Chebyshev parameters to use as follows. Let κ be the total number of Newton steps to be executed. Of course κ is not known; however, for the moment, assume that it is. Let c_{m-1} , $m = k + 1, \dots, l$, $l \leq \kappa$ be the reciprocals of the roots of the polynomial

$$C_{l-k}(\lambda) = T_{l-k} \left(\frac{d - \lambda}{c_{\text{ellipse}}} \right)$$

where T_j is the Chebyshev polynomial of degree j .

In place of (17), execute

$$y_n^{(m)} = y_n^{(m-1)} + c_{m-1} r^{(m-1)}, \quad (22)$$

for $m = k + 1, \dots, l$.

There is an obvious difficulty to resolve: the number l must be determined in advance, and must satisfy $l \leq \kappa$, where κ is unknown. One approach is this. Choose some fixed l and compute the Chebyshev parameters. It is not expected that $l = \kappa$ so that either there are too few parameters c_m to get to Newton step κ or there are too many. If there are not enough, then recycle the old, computed parameters c_m , in which case eventually either there will be unused parameters c_m or all parameters will have been used. Thus, plans must be made for unused parameters. To prepare for too many parameters, order c_k, \dots, c_{l-1} by either of the methods of [26] or [8], which exploit the known mapping of Chebyshev polynomial roots onto the unit circle to obtain an ordering for which, heuristically, $\|r^{(m)}\| < \|r^{(m-1)}\|$. (These ordering techniques are more general, however, and are not restricted to the Chebyshev case.) With this ordering one may reasonably choose $l - k = 4$, even though this number of parameters is not likely to be reached within a single time step by a solver such as DASSL.

Another approach to the problem of excessive parameters c_m is to use the second order formulation of the Chebyshev iteration. The following is an adaptation of the algorithm in [17]. Recall that the residual $r^{(k)}$ is $F(y_n^{(k)})$.

1. Set $\alpha_{k+1} := 1/d$.
2. Set $\Delta y_n^{(k)} := \alpha_{k+1} r^{(k)}$.
3. Set $y_n^{(k+1)} := y_n^{(k)} + \Delta y_n^{(k)}$.
4. Set $r^{(k+1)} := F(y_n^{(k+1)})$.
5. Do for $m = k + 2$ by 1 until convergence:
 - 5.1 Set $\alpha_m := 1/([d - c_{\text{ellip}}^2]\alpha_{m-1}/4)$.
 - 5.2 Set $\gamma_m := d\alpha_m - 1$.
 - 5.3 Set $\Delta y_n^{(m-1)} := \gamma_m \Delta y_n^{(m-2)} + \alpha_m r^{(m-1)}$.
 - 5.4 Set $y_n^{(m)} := y_n^{(m-1)} + \Delta y_n^{(m-1)}$.
 - 5.5 Set $r^{(m)} := F(y_n^{(m)})$.

Enddo

For another application of Richardson's method to nonlinear problems, see [10, 12].

5.3 Changing the Stepsize

Changing the stepsize causes a potentially large change in the Jacobian of F . The effect of this on the Newton's method Chebyshev parameters will be estimated under the assumption that no preconditioning was used at step n to solve the linear systems with matrix \hat{G} . Additional assumptions are (i) the matrix \hat{G} does not change; (ii) the solution of the *linear* systems at step n with matrix \hat{G} has been by the Manteuffel adaptive Chebyshev algorithm, which yields the convex hull of \hat{G} ; (iii) \hat{G} was not preconditioned during the Chebyshev iteration; and (iv) that the eigenvectors of $F'(y_{n+1}^{(0)})$ and $F'(y_{n'}^{(m')})$ are approximately the same.

5.3.1 Operator Approximation at Step $n + 1$

We outline a technique to apply the work to compute the parameters at step n to the computation of the parameters at step $n + 1$.

At this point it is convenient to denote the dependence of F and M on n by F_n and M_n .

The acceleration parameters c_m are determined by the convex hull of approximations to

eigenvalues of

$$M_{n+1} = \hat{G}^{-1} F'_{n+1}(y_{n+1}^{(0)}). \quad (23)$$

Consider the expression on the right. Recall that

$$\hat{G} = h_{n'}^{-1} I - f'(y_{n'}^{(m')}).$$

The other factor on the right of (23) is

$$F'_{n+1}(y_{n+1}^{(0)}) = h_{n+1}^{-1} I - f'(y_{n+1}^{(0)}).$$

Therefore,

$$M_{n+1} = [h_{n'}^{-1} I - f'(y_{n'}^{(m')})]^{-1} [h_{n+1}^{-1} I - f'(y_{n+1}^{(0)})]. \quad (24)$$

5.3.2 The Convex Hull at Step $n + 1$

The convex hull of M_{n+1} in (24) may be estimated in terms of known quantities as follows.

It is assumed that

$$f'(y_{n+1}^{(0)}) \approx f'(y_n^{(0)}). \quad (25)$$

Let

$$\mu_n = \frac{1}{h_{n+1}} - \frac{1}{h_n}. \quad (26)$$

From (25) and (26), (24) becomes

$$M_{n+1} \approx [h_n^{-1} I - f'(y_n^{(m')})]^{-1} [\mu_n I + h_n^{-1} I - f'(y_n^{(0)})] \quad (27)$$

$$= \mu_n \hat{G}^{-1} + M_n. \quad (28)$$

The convex hull of M_{n+1} is needed. The eigenvectors of $f'(y_n^{(0)})$ and $f'(y_n^{(m')})$ are assumed to be approximately the same. It follows that the eigenvectors of M_n and \hat{G} are approximately the same. Therefore, if the convex hull of each matrix on the right of (28) were known then an approximate convex hull of the sum could be easily computed. The convex hull of M_n is assumed known from the preceding time step. It remains to determine the convex hull of \hat{G}^{-1} approximately.

The adaptive Chebyshev method was assumed to have been used to solve the *linear* systems at the preceding step, n , for which the matrix is \hat{G} .⁴ The convex hull of \hat{G} is therefore known. An approximation to the convex hull of \hat{G}^{-1} is therefore easily obtained.

⁴Note that this application of the adaptive Chebyshev method is in the execution of the inner iteration rather than the outer Newton iteration.

6 Summary

In this paper, a simplified analysis has been presented for the predictor error. It suggests that when using an iterative method for the inner iteration (with Newton's method or a variant of Newton's method as the outer iteration), the iterative method should be one for which components of the error are damped approximately uniformly. Two classes of iterative methods are sketched for which the error is damped in a uniform way, the adaptive Chebyshev method of Manteuffel and Richardson's method. Richardson's method is preferable if the spectrum of the inner iteration matrix is not well approximated by an ellipse.

If the (modified) Newton's method step is viewed as one step of Richardson's method, the usual techniques for computing iteration parameters adaptively may be used to enhance convergence of Newton's method with a parameter sequence. In the case when no preconditioning is used for the inner iteration, the parameters may be recomputed at low cost when the stepsize changes.

References

- [1] P. N. Brown and A. C. Hindmarsh. *Matrix-free methods for stiff systems of ODEs*. **SIAM J. Numer. Anal.**, Vol. 23, pp. 610–638, 1986.
- [2] P. N. Brown and Youcef Saad. *Hybrid Krylov methods for Nonlinear Systems of Equations*. **SIAM J. Sci. Statist. Comput.**, Vol. 11, pp. 450–481, 1990.
- [3] T. F. Chan and K. R. Jackson. *The use of iterative linear equation solvers for large systems of stiff IVPs for ODEs*. **SIAM J. Sci. Statist. Comput.**, Vol. 7, pp. 378–417, 1986.
- [4] A. T. Chronopoulos. “Nonlinear CG-Like Iterative Methods”, Computer Science Department Technical Report TR 89-2, University of Minnesota, Minneapolis, 1989.
- [5] A. C. Curtis. *Stiff ODE initial value problems and their solution*. In: **The State of the Art in Numerical Analysis**, A. Iserles and M. J. D. Powell, eds., Oxford University Press, New York, pp. 433–450, 1987.
- [6] H. C. Elman, Y. Saad and P. E. Saylor. *A hybrid Chebyshev Krylov subspace algorithm for solving nonsymmetric systems of linear equations*. **SIAM J. Sci. Statist. Comput.**, Vol. 6, pp. 840–855, 1985.
- [7] M. Eiermann, R. S. Varga and W. Niethammer. *Iterationsverfahren für nichtsymmetrische Gleichungssysteme und Approximationsmethoden im Komplexen*. **Jber. d. Dt. Math.-Verein**, Vol. 89, pp. 1–32, 1987.
- [8] B. Fischer and L. Reichel. *A stable Richardson iteration method for complex linear systems*. **Numer. Math.**, to appear.
- [9] C. W. Gear and Y. Saad. *Iterative solution of linear equations in ODE codes*. **SIAM J. Sci. Statist. Comput.**, Vol. 5, 1984.
- [10] G. H. Golub and R. Kannan. *Convergence of a two-stage Richardson process for nonlinear equations*. **BIT** 26, 209–216, 1986.
- [11] G. H. Golub and J. H. Welsch. *Calculation of Gaussian quadrature rules*. **Math. Comput.**, Vol. 23, 221–230, 1969.

- [12] M. H. Gutknecht. *Stationary and almost stationary iterative (k, l) -step methods for linear and nonlinear systems of equations*. **Numer. Math.**, Vol. 56, 179-213, 1989.
- [13] L. A. Hageman and D. M. Young. **Applied Iterative Methods**, Academic Press, New York, 1981.
- [14] C. Johnson. *Error estimates and adaptive time-step control for a class of one-step methods for stiff ordinary differential equations*. **SIAM J. Numer. Anal.**, Vol. 25, pp. 908-926, 1988.
- [15] T. Kerkhoven and Y. Saad. "Acceleration Techniques for Decoupling Algorithms in Semiconductor Simulation", Department of Computer Science Technical Report No. UIUCDCS-R-87-1363, Urbana, 1987.
- [16] F. T. Krogh and K. Stewart. *Asymptotic absolute stability $(h \rightarrow \infty)$ for BDFs applied to stiff differential equations*. **ACM Trans. Math. Software**, Vol. 10, pp. 45-56, 1984.
- [17] T. A. Manteuffel. *Adaptive procedure for estimating parameters for the nonsymmetric Tchebyshev iteration*. **Numer. Math.**, Vol. 31, pp. 183-208, 1978.
- [18] W. L. Miranker and I-L. Chern. "Dichotomy and Conjugate Gradients in the Stiff Initial Value Problem", Tech. Rep. RC 8032, IBM, Thomas J. Watson Research Center, 1980.
- [19] G. Opfer and G. Schober. *Richardson's iteration for nonsymmetric matrices*. **Linear Alg. Applic.**, Vol. 58, pp. 343-361, 1984.
- [20] K. E. Brenan, S. L. Campbell and L. R. Petzold. **Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations**, North-Holland, New York, 1989.
- [21] Y. Saad and M. H. Schultz. *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*. **SIAM J. Sci. Statist. Comput.**, Vol. 7, pp. 856-869, 1986.
- [22] P. E. Saylor. *Leapfrog variants of iterative methods for linear algebraic equations*. **J. Comput. Appl. Math.**, Vol. 24, pp. 169-193, 1988.

- [23] P. E. Saylor. *An adaptive algorithm for Richardson's method*. In: **Iterative Methods for Large Linear Systems**, Academic Press, New York, 1990.
- [24] P. E. Saylor and D. C. Smolarski. *Computing the roots of complex orthogonal and kernel polynomials*. **SIAM J. Sci. Statist. Comput.**, Vol. 9, 1988.
- [25] L. F. Shampine. *Evaluation of implicit formulas for the solution of ODEs*. **BIT**, Vol. 19, pp. 495-502, 1979.
- [26] H. Tal-Ezer. "Polynomial Approximation of Functions of Matrices and Applications", ICASE Report No. 87-63, Institute for Computer Applications in Science and Engineering, NASA Langley Research Center, Hampton, Va., 1987.



Report Documentation Page

1. Report No. NASA CR-182074 ICASE Report No. 90-51	2. Government Accession No.	3. Recipient's Catalog No.	
4. Title and Subtitle LINEAR ITERATIVE SOLVERS FOR IMPLICIT ODE METHODS		5. Report Date August 1990	
		6. Performing Organization Code	
7. Author(s) Paul E. Saylor Robert D. Skeel		8. Performing Organization Report No. 90-51	
		10. Work Unit No. 505-90-21-01	
9. Performing Organization Name and Address Institute for Computer Applications in Science and Engineering Mail Stop 132C, NASA Langley Research Center Hampton, VA 23665-5225		11. Contract or Grant No. NAS1-18605	
		13. Type of Report and Period Covered Contractor Report	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Langley Research Center Hampton, VA 23665-5225		14. Sponsoring Agency Code	
15. Supplementary Notes Langley Technical Monitor: Richard W. Barnwell Proc. of the Copper Mtn. Meeting on Iterative Methods, April 1990. Sub- mitted to SIAM Journal of Statistical and Scientific Computing.			
<u>Final Report</u>			
16. Abstract In this paper we consider the numerical solution of stiff initial value problems, which lead to the problem of solving large systems of mildly nonlinear equations. For many problems derived from engineering and science, a solution is possible only with methods derived from iterative linear equation solvers. A common approach to solving the nonlinear equations is to employ an approximate solution obtained from an explicit method. In this paper we shall examine the error to determine how it is distributed among the stiff and non-stiff components, which bears on the choice of an iterative method. Our conclusion is that error is (roughly) uniformly distributed, a fact that suggests the Chebyshev method (and the accompanying Manteuffel adaptive parameter algorithm). We describe this method, also commenting on Richardson's method and its advantages for large problems. We then apply Richardson's method and the Chebyshev method with the Manteuffel algorithm to the solution of the nonlinear equations by Newton's method.			
17. Key Words (Suggested by Author(s)) stiff ODEs, stiff systems, implicit difference methods, Chebyshev method, Richardson's method		18. Distribution Statement 64 - Numerical Analysis Unclassified - Unlimited	
19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 21	22. Price A03